

1 Dynamic languages symposium chair's welcome: Hardware tansactional memory

support for lightweight dynamic language evolution

Nicholas Riley, Craig Zilles

October 2006 Companion to the 21st ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications OOPSLA '06

Publisher: ACM Press

Full text available: pdf(325.22 KB) Additional Information: full citation, abstract, references, index terms

Lightweight dynamic language runtimes have become popular in part because they simply integrate with a wide range of native code libraries and embedding applications. However, further development of these runtimes in the areas of concurrency, efficiency and safety is impeded by the desire to maintain their native code interfaces, even at a source level. Native extension modules' lack of thread safety is a significant barrier to dynamic languages' effective deployment on current and future multic ...

**Keywords:** Python, concurrency, dynamic languages, locking, safety, transactional memory

Model-carrying code: a practical approach for safe execution of untrusted
 applications



applications

R. Sekar, V.N. Venkatakrishnan, Samik Basu, Sandeep Bhatkar, Daniel C. DuVarney
October 2003 ACM SIGOPS Operating Systems Review, Proceedings of the nineteenth
ACM symposium on Operating systems principles SOSP '03, Volume 37 Issue

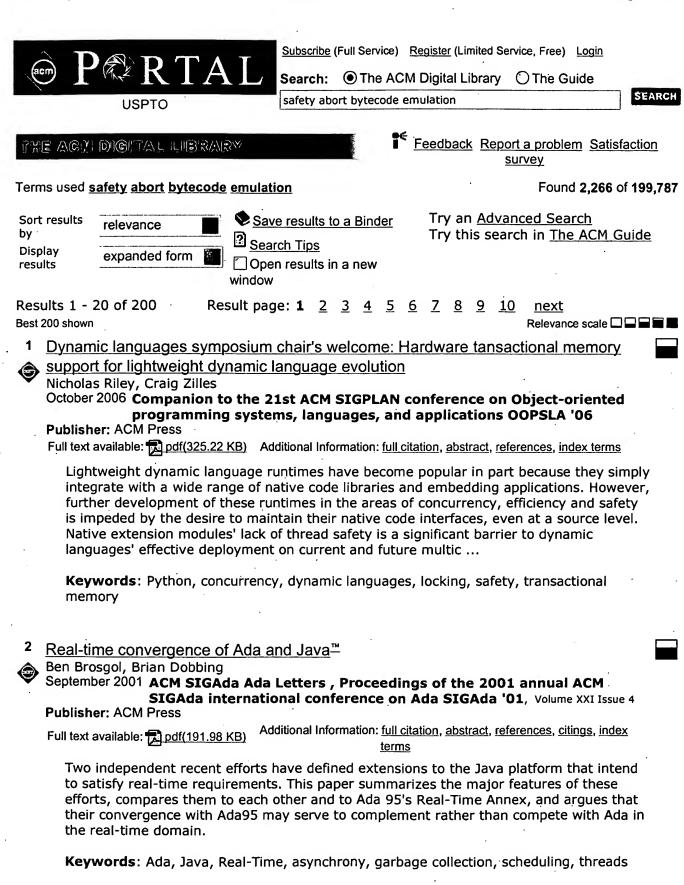
Publisher: ACM Press

Full text available: pdf(301.30 KB)

Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index</u> terms

This paper presents a new approach called *model-carrying code* (MCC) for safe execution of untrusted code. At the heart of MCC is the idea that untrusted code comes equipped with a concise high-level model of its security-relevant behavior. This model helps bridge the gap between high-level security policies and low-level binary code, thereby enabling analyses which would otherwise be impractical. For instance, users can use a fully automated verification procedure to determine if the code ...

Keywords: mobile code security, policy enforcement, sand-boxing, security policies



http://portal.acm.org/results.cfm?coll=ACM&dl=ACM&CFID=19287689&CFTOKEN=59... 4/6

JRes: a resource accounting interface for Java

Grzegorz Czajkowski, Thorsten von Eicken